Machine Learning Homework

硕-8089 3118111009 许林松

I. QA

1

What is the likelihood ratio

$$\frac{p(x|C_1)}{p(x|C_2)} \tag{1}$$

in the case of Gaussian densities?

SOLUTION:

$$\frac{p(x|C_1)}{p(x|C_2)} = \frac{\frac{1}{\sqrt{2\pi}}\sigma_1 exp[-\frac{(x-\mu_1)^2}{2\sigma_1^2}]}{\frac{1}{\sqrt{2\pi}}\sigma_2 exp[-\frac{(x-\mu_2)^2}{2\sigma_2^2}]} \tag{2}$$

If we have $\sigma_1^2 = \sigma_2^2 = \sigma^2$, we can simplify

$$\frac{p(x|C_1)}{p(x|C_2)} = exp[-\frac{(x-\mu_1)^2}{2\sigma^2} + \frac{(x-\mu_2)^2}{2\sigma^2}] \tag{3}$$

$$= exp[\frac{(\mu_1-\mu_2)}{\sigma^2}x + \frac{(\mu_2^2-\mu_1^2)}{2\sigma^2}] \tag{4}$$

$$= exp(wx + w_0) \tag{5}$$

for $w = (\mu_1 - \mu_2)/\sigma^2$ and $w_0 = (\mu_2^2 - \mu_1^2)/2\sigma^2$

2

In regression we saw that fitting a quadratic is equivalent to fitting a linear model with an extra input corresponding to the square of the input. Can we also do this in classification?

SOLUTION: Yes. We can define new, auxiliary variables corresponding to powers and cross-product terms and then use a linear model.

For example, let us say we have two variables x_1 and x_2 and we want to make a quadratic fit using them, namely,

$$f(x_1, x_2) = w_0 + w_1 z_1 + w_2 z_2 + w_3 x_1 x_2 + w_4 (x_1)^2 + w_5 (x_2)^2$$

we can define $z_1 = x_1$,$z_2 = x_2$,$z_3 = x_1 x_2$,$z_4 = (x_1)^2$ and $z_5 = (x_2)^2$ and then use a linear model to learn $w_i, i = 0, \ldots, 5$. The linear discriminant in the five-dimensional $(z_1, z_2, z_3, z_4, z_5)$ space corresponds to a quadratic discriminant in the two-dimensional (x_1, x_2) space.

3

Let $P(Y = c_k) = \theta_k$, where$k \in \{1, 2 \ldots k\}$

So we can get:

$$P(Y) = \sum_{k=1}^{K} \theta_k I(Y = c_k) \tag{6}$$

where $I = 1 \quad if \quad Y = c_k, \quad else \quad I = 0$

Note that,

$$L(\theta_k; y_1, y_2 \ldots y_N) = \Pi_{i=1}^{N} P(y_i) = \Pi_{k=1}^{K} \theta_k^{N_k} \tag{7}$$

where N is the total number of samples, N_k is the number of samples which meet the condition $Y = c_k$ Then we can get log function

$$l(\theta_k) = ln(L(\theta)) = \sum_{k=1}^{K} N_k ln\theta_k \tag{8}$$

We want to maximum it, and with the constraints $\sum_{k=1}^{K} \theta_k = 1$ so we can use Lagrange multipliers to solve this problem. We get

$$l(\theta_k, \lambda) = \sum_{k=1}^{K} N_k ln\theta_k + \lambda(\sum_{k=1}^{K} \theta_k - 1) \tag{9}$$

then we can get the equation as below

$$\frac{N_k}{\theta_k} + \lambda = 0 \tag{10}$$

because that $\sum_{k=1}^{K} \theta_k = 1$ and $\sum_{i=1}^{k} N_i = N$, so we get

$$\theta_k = \frac{N_k}{N} \tag{11}$$

II. MATH

1

$$P(cancer|+) = \frac{P(cancer,+)}{P(+)} \tag{12}$$

$$= \frac{P(cancer)P(+|cancer)}{P(cancer)P(+|cancer) + P(not)P(+|not)} \tag{13}$$

$$= \frac{0.01 \times 0.9}{0.01 \times 0.9 + 0.99 \times 0.1} \tag{14}$$

$$= 0.083 \tag{15}$$

III. CODING

I use the package named "jieba" to cut the Chinese words and then compute the $tf - idf$, finally feed them to a logistic regression classifier. I get the result as below:

Console:

Building prefix dict from the default dictionary ...

Loading model from cache /var/***

Loading model cost 0.608 seconds.

Prefix dict has been built succesfully.

length of corpus is: 3312

how many words: 24592

tf-idf shape: (3312,24592)

val mean accuracy: 0.8295042321644498

	precision	recall	f1-score	support
0	0.91	0.94	0.92	239
1	0.95	0.64	0.76	192
2	1.00	0.10	0.17	63
3	0.74	0.99	0.85	336
avg/total	0.86	0.83	0.80	830

TABLE I

PREDICTION RESULT

You can find my project and code from here